Design and Implementation of a Context-Aware Decision Algorithm for Heterogeneous Networks

Tansir Ahmed
BenQ Mobile, MD PBM TI 8
Haidenauplatz 1
81667 Munich, Germany
+49 89 722 63294
Tansir.Ahmed@BenQ.com

Kyandoghere Kyamakya
ISYS, Universität Klagenfurt
Universitätstrasse 65
9020 Klagenfurt, Austria
+43 463 2700 3540
kyamakya@isys.uni-klu.ac.at

Markus Ludwig
BenQ Mobile, MD PBM TI 4
Haidenauplatz 1
81667 Munich, Germany
+49 89 722 49250
Markus.Ludwig@BenQ.com

ABSTRACT
Wireless networks and mobile terminals are evolving towards being heterogeneous. In this environment, intelligent handover decision, beyond traditional ones that are based on only signal strength, is needed so that terminals can select the best option available from diverse networks and services as per user requirements. In the process, it would enable user applications to switch automatically between active interfaces that best suit them based on application requirements and interface capabilities and to use multiple radio interfaces simultaneously ensuring the optimum usage of the network resources available to the terminal. To fulfill the above requirements, this paper proposes the design and implementation of a context-aware vertical handover decision algorithm suitable for multimode mobile devices in heterogeneous networks based on the *Analytic Hierarchy Process (AHP)*.

Categories and Subject Descriptors
D.3.3 [**Computing Methodologies**]: Algorithms – *algebraic algorithms.*

General Terms
Algorithms, Design.

Keywords
Context model, Context-awareness, Heterogeneous networks, Vertical handover decision algorithm.

1. INTRODUCTION
Present day wireless communications networks and devices are experiencing a paradigm shift. Rapid emergence of diverse access technologies, e.g. WLAN, Bluetooth, 3GPP cellular networks (GSM, GPRS, and UMTS), DVB-H, etc would result in evolution of wireless networks towards heterogeneous all-IP infrastructure. In this heterogeneous *overlaying* infrastructure, users should be given the freedom to roam globally among multitude points of attachment of different access networks (*vertical handover*) as per their service requirements. Conventional single interface mobile terminals are also evolving into multimode terminals. Currently, these multimode terminals do not possess true multimode functionality. They are limited to use only one radio interface at a time. But in the given heterogeneous scenario, these terminals should have true multimode functionality that would enable user applications to switch automatically between active interfaces that best suit them based on application requirements and interface capabilities and to use multiple radio interfaces simultaneously. This would definitely optimize the usage of the network resources available to the device. Traditional *horizontal handover (HO)* decision mechanisms that mainly depend on signal strength for decision making are unable to realize the above requirements.

In the given circumstances, we have developed an intelligent HO decision algorithm including the session transfer, which takes into account, as much as possible, intelligence residing on the terminal side as well as on the network side, collectively known as *context information*. The simple design of the algorithm would make it suitable for practical multimode mobile devices (e.g. PDA) that have several capability constraints like processor speed, memory size, power consumption, etc. On the contrary, it is versatile enough to be configurable by users. The algorithm is based on the *Analytic Hierarchy Process (AHP)* [1].

The rest of the paper is organized as follows. Section 2 highlights the related work. Section 3 and 4, respectively, illustrate the design and the implementation of the algorithm. Section 5 briefly discusses the validity of the algorithm based on test runs in wireless environment. Finally, section 6 concludes the paper.

2. RELATED WORK
Hongyan *et al.* [2] and Chan *et al.* [3] propose a *fuzzy* based multiple-criteria decision making process to perform access network selection and vertical HO based on the cost constraints and application priorities specified by users. Stemm *et al.* [4] and Pahlavan *et al.* [5] describe intelligent HO procedures especially for hybrid networks considering the type of the radio access technology and the signal strength. In [6], different HO policies for heterogeneous networks are used considering as HO decision parameters mainly the type of air interface and the available bandwidth at the *access router (AR)*.

The above mentioned methods either consider only a few context parameters or are too complex to be suitable for practical multimode mobile devices that possess limited resources.

SAC'06, April, 23-27, 2006, Dijon, France.

Balasubramaniam *et al.* [7] also uses the AHP method in their decision making process. However, it lacks an elaborate model that would consider a wide variety of the most important contexts and their grouping, precise calculation methods for mapping relevant contexts in the chosen model, user interactions in the process, and lastly, the session transfer based on the decision.

3. THE DECISION ALGORITHM

The task of our context-aware decision algorithm is to select the most suitable interface for a given application among multiple options that would satisfy some primary objectives based on the values of some context parameters. In this regard, the AHP model [1], which is a well-known and proven mathematical process to identify the most suitable choice among multiple alternatives based on some predefined objectives, perfectly fits into our decision making process. We have considered *mobile-initiated and controlled* vertical HO for the decision algorithm.

3.1 Context Model

The context model chosen for the decision algorithm is shown in Table 1. The contexts that do not change very often are *static* context information, whereas those that change quite frequently and may loose accuracy over time are *dynamic* context information.

Table 1. Context model for decision algorithm

Context Type	Terminal Side	Network Side
Static	Device capabilities, service types, QoS requirements of services, user preferences	Provider's profile
Dynamic	Running application type, reachable access points	Current QoS parameters of AP

On the terminal side, *device capabilities* include display size, resolution, battery life, memory, processor speed, and available interfaces. All services offered by a terminal are classified into three *service types*, namely conversational/real-time services, interactive services, and streaming services, where each of them has its own *QoS requirements*. *User preferences* are grouped as *interface preferences* for multimode terminal and *service preferences* (precedence of service types, expected QoS, and cost constraints). *Running application types* defines the service profile of currently running applications. *Reachable access points (APs)* identifies currently available networks and addresses of the APs.

On the network side, *Service provider's profiles* consist of provider's identity and charging models. *Current QoS parameters* define the current status of the available network QoS parameters.

3.2 Architecture of the Decision Algorithm

The architecture of the decision algorithm is shown in Figure 1. In the model, a user defines his preferences in some categories that should meet both application requirements and device capabilities. Capabilities of available networks are discovered and compared with the defined preferences by employing the decision algorithm, and the most suitable network corresponding to the preferences is selected. Finally, applications that need to be transferred to the selected interface are switched. The decision algorithm is processed for each service type currently running in the device.

In accordance with the AHP method, at first, we have to define some *primary objectives* for our decision algorithm taking into account the preferences likely to be the most interesting to users (e.g. cost, expected QoS, interface priority based on coverage, etc) and 3GPP defined *Quality of Service (QoS)* parameters [8]. We have chosen the following *six* primary objectives:

1) Consider interface priority.
2) Minimize cost.
3) Maximize mean throughput.
4) Minimize delay.
5) Minimize jitter.
6) Minimize Bit Error Rate (BER)/Frame Error Rate (FER).

3.2.1 Pre-configuration

3.2.1.1 Stage 1: Taking User Inputs

For any of the three service types, a user needs to define *three* sets of relative priorities. These three sets are (i) relative priorities among primary objectives (*objective priorities*) (ii) relative priorities among available interfaces in a device (*interface priorities*) and (iii) relative priorities among three types of services (*application priorities*). User preferences are taken as *discrete* values or *scores*. However, in order to make the model more user-friendly available options, in each case, are labeled with suitable *literals*. The user only needs to arrange the literals in a descending order starting with the one with the highest priority. Based on the arrangement of the literals priority scores between 1 and 9 are assigned automatically at the backend, where 1 denotes the most preferred one and 9 denotes the least preferred one. Priority scores are equal-spaced integers whose space-gap is defined by (1), where N_p denotes the number of parameters, L_u and L_l denote the highest and lowest possible scores i.e. 9 and 1, respectively, and G denotes the numeric space-gap between two subsequent scores, which is rounded off to the nearest integer.

$$G = \frac{L_u - L_l}{N_p} \qquad (1)$$

As an example, among the primary objectives mentioned earlier objective 1 is labeled as "Desired Interface", objective 2 as "Lowest Cost", and objectives 3 to 6, in a group, as "Best Quality". Here, (1) results in $G = 3$ while using $L_u = 9$, $L_l = 1$, and $N_p = 3$. If a user arranges the literals as in the order "Lowest Cost", "Best Quality", and "Desired Interface" objective 2, objectives 3-6, and objective 1 have scores of 1, 4, and 7, respectively. Since "Best Quality" is the group of four parameters objectives 3-6 have the same score, 4. Similar measures are taken in case of interface and application priorities. For the former, N_p equals to the number of interfaces in the terminal and for the latter, types of services. Each of the sets of literals includes a "Default" option.

3.2.1.2 Mapping Limit Values

Some context information especially network QoS parameters are very dynamic. Therefore, it makes sense to express QoS preferences from users as *continuous* values or *limits* in order to provide better flexibility while comparing them with the network QoS parameters.

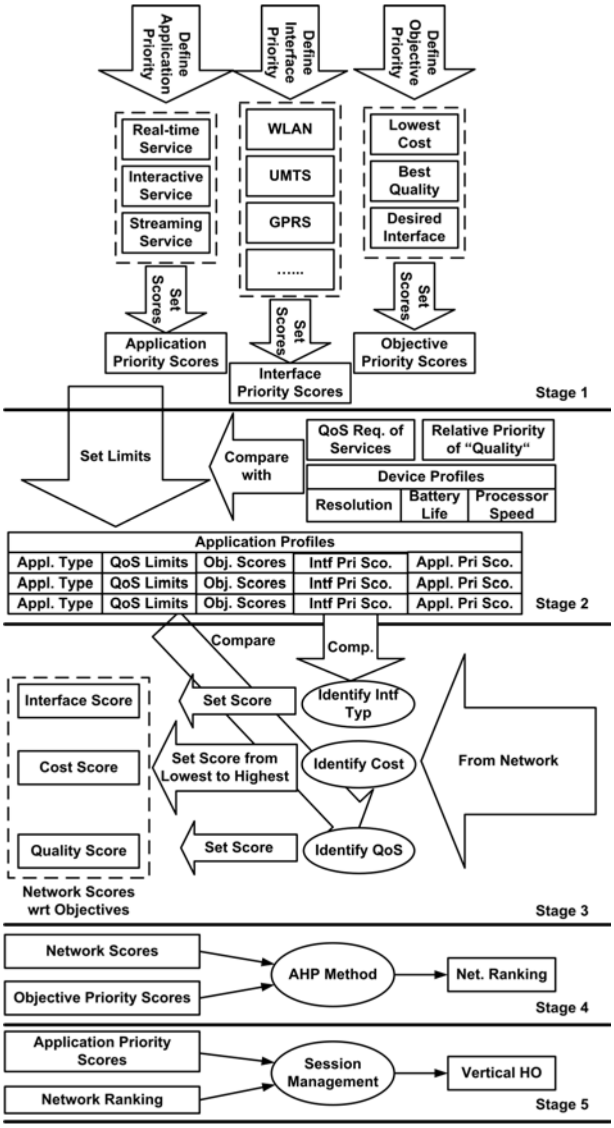

Figure 1. Architecture of the decision algorithm

At this stage, suitable limit values (upper and lower) for the four QoS parameters related directly to objectives 3-6 are *mapped* at the backend for each of the three service types. While fixing the limit values, it is important to note that high values are not always better for all the four QoS parameters. It is always preferable to have values as high as possible for mean throughput, whereas as low as possible for delay, jitter, and BER/FER. In case of mean throughput, the lower limit is always a fixed value, i.e. *minimum requirement* (e.g. ≥ 4 kbps for conversational service [8]). This value is based on the contexts like QoS requirements of specific service type and device capabilities. The upper limit varies in accordance with the objective priority scores of the QoS based objectives (objectives 3-6) set earlier. For example, if the objectives 3-6 have the highest priority (priority score equals 1) the upper limit is set at the highest possible value (e.g. > 25 kbps for conversational service [8]), on the contrary, if they have the lowest priority (priority score equals 7) it is set much nearer to the lower limit (e.g. 8 kbps for conversational service). The limit values for the other three QoS parameters are fixed likewise, except that the upper limit is always a fixed value in this case, i.e. *maximum tolerance* (e.g. ≤ 400 ms one-way delay for conversational service [8]) and the lower limit varies according to the objective priority scores.

At the end, we have three sets of *preconfigured* data (scores and limits) for the three service types. They are grouped together and stored as the *application profiles* (see Figure 1) where individual service type is identified by the *application type*.

3.2.2 Real-Time Calculations

The following stages perform real-time calculations for a particular type of running application.

3.2.2.1 Stage 3: Assigning Scores to Networks

At this stage, capabilities of the reachable networks (including the current network, if any) are compared with the preconfigured user preferences (scores and limits based on the six objectives) and suitable scores are assigned to each of the networks. A multimode mobile device would always monitor (layer-2 or layer-3 monitoring, or both) each of its interfaces for reachable networks. It is assumed that shared contexts of networks like current QoS parameters and cost would always be advertised by the available networks, or the terminal may utilize layer-2 or layer-3 probing.

Assignment of scores to the available networks based on discrete preferences like interface priority and cost constraint is straightforward. The same interface priority score, already defined by the user in stage 1, is assigned to the available network depending on its type. In case of cost objective, all the available networks are compared with each other and assigned with appropriate equal-spaced scores between 1 and 9 based on (1) in a descending order, where the cheapest network has a score of 1. If a particular network does not advertise the cost information it is assigned with a score of 9 (costliest network) as a default value.

$$\begin{aligned} S_i &= \left(1-\frac{n_i-l_i}{u_i-l_i}\right)\times 10 && ;l_i < n_i < u_i \\ &= 1 && ;n_i \ge u_i \\ &= 9 && ;n_i \le l_i \end{aligned} \quad (2)$$

$$\begin{aligned} S_i &= \left(\frac{n_i-l_i}{u_i-l_i}\right)\times 10 && ;l_i < n_i < u_i \\ &= 1 && ;n_i \le l_i \\ &= 9 && ;n_i \ge u_i \end{aligned} \quad (3)$$

In case of continuous preferences, QoS parameters of all available networks are compared with the individual parameter limit values defined in stage 2. If u_i and l_i denote the upper and lower limits of a particular continuous preference and n_i denotes the value offered by a network for that particular parameter the network score, S_i, based on the preference is calculated using (2) and (3). Eq. (2) is used for continuous preferences like mean throughput, where the target value is preferred to be as high as possible. On the contrary, (3) is used for continuous preferences like delay, jitter, and BER/FER, where the target value is preferred to be as low as possible. If there is any missing parameter i.e. not advertised by a particular network its default value is used. Values of l_i and u_i are the default values for (2) and (3), respectively.

3.2.2.2 Stage 4: Calculating Network Ranking

At this stage, *ranking* of the available networks are performed based on the objective priority scores and network scores assigned at stage 1 and 3, respectively. The calculations use the AHP method, which is a three step process [1].

Step 1: At first, the relative scores among the objective priority scores set by the user at stage 1 are calculated. Relative scores are scaled linearly between 1 and 9 [1]. Relative scores between any two particular scores are calculated using (4), (5), and (6), where RS_{ab} is the relative score between parameters a and b, and S_a and S_b are their respective scores.

$$\frac{1}{RS_{ab}} = \left(1 - \frac{S_b}{S_a}\right) \times 10 \quad ; S_a > S_b \tag{4}$$

$$RS_{ab} = \left(1 - \frac{S_a}{S_b}\right) \times 10 \quad ; S_a < S_b \tag{5}$$

$$RS_{ab} = 1 \quad ; S_a = S_b \tag{6}$$

With the calculated relative scores the priorities (i.e. weights) for the six objectives in terms of the overall goal i.e. selecting a suitable network are calculated using *pairwise comparison matrix* [1] for objectives. It consists of the relative scores calculated in the previous step. The dimension of the pairwise comparison matrix A for the objectives, as shown in (7), is flexible and dependent on the number of chosen objectives (6 × 6, in our case). Matrix A is then normalized by dividing each element by individual sum of column. The normalized matrix A_{norm} is shown in (8). At the end, the average values of each row for objective i are calculated to give the priorities for each objective (p_1, p_2, p_3, p_4, p_5, p_6) with respect to the overall goal using (9).

$$A = \begin{bmatrix} 1 & RS_{12} & RS_{13} & RS_{14} & RS_{15} & RS_{16} \\ \frac{1}{RS_{12}} & 1 & RS_{23} & RS_{24} & RS_{25} & RS_{26} \\ \frac{1}{RS_{13}} & \frac{1}{RS_{23}} & 1 & RS_{34} & RS_{35} & RS_{36} \\ \frac{1}{RS_{14}} & \frac{1}{RS_{24}} & \frac{1}{RS_{34}} & 1 & RS_{45} & RS_{46} \\ \frac{1}{RS_{15}} & \frac{1}{RS_{25}} & \frac{1}{RS_{35}} & \frac{1}{RS_{45}} & 1 & RS_{56} \\ \frac{1}{RS_{16}} & \frac{1}{RS_{26}} & \frac{1}{RS_{36}} & \frac{1}{RS_{46}} & \frac{1}{RS_{56}} & 1 \end{bmatrix} \tag{7}$$

$$A_{norm} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} \end{bmatrix} \tag{8}$$

$$p_i = \frac{b_{i1} + b_{i2} + b_{i3} + b_{i4} + b_{i5} + b_{i6}}{6} \tag{9}$$

Step 2: The relative scores among the scores of the available networks assigned at stage 3 in terms of individual objective are calculated using (4), (5), and (6). Then the network conformances (i.e. weights), c_{ij}, for i number of available networks in terms of each of j number objectives are calculated in similar fashion as described in step 1.

Step 3: The overall ranking of each available network is determined by calculating the sum of products of network conformances in terms of individual objective (obtained from step 2) and objective priorities for that particular objective (obtained from step 1). For i number of available networks and j number of objectives, the overall ranking R_i can be obtained from (10). R_i is always in the range of 0-1. The network with the highest rank is finally selected.

$$R_i = \sum_1^{ij} c_{ij}(p_j) \tag{10}$$

3.2.2.3 Stage 5: Session Management

At this final stage, an efficient session transfer scheduling algorithm is employed in order to switch applications to the selected network. The scheduling algorithm takes into account the application priority score set by the user at stage 1 and the rank of the selected network obtained from (10) at stage 4. For i number of running applications the overall score, O_i, is calculated using (11), where, R_d and R_i, respectively, are the ranks of the current and the selected network for the i^{th} application, and a'_i is the normalized value of its application priority score, a_i, given by (12).

$$O_i = a'_i (R_i - R_d) \tag{11}$$

$$a'_i = \left(1 - \frac{a_i}{10}\right) \tag{12}$$

The value of O_i is always between -0.9 to +0.9. For a given application, $O_i = 0$ or $O_i < 0$ means that the application is already using the optimum interface and it needs not to be switched to an alternative one. For all $O_i > 0$, applications are switched in accordance with their O_is in a descending order starting with the one with the highest O_i.

4. IMPLEMENTATION

The algorithm is implemented in the decision module of the *Reference Architecture* briefly described here. The target device for the implementation has been selected as *MDA III* from *T-Mobile* (Intel PXA-263 400 MHz processor, 128 MB RAM, 96 MB ROM, GSM Quad-band, multimode PDA) on *Windows Mobile 2003 SE* platform, which should be updated to *Windows CE 5.0* at some later stage. Basic modules of the Reference Architecture are shown in Figure 2.

The *Light Network Capability Discovery (LNCD)* module periodically monitors all interfaces and discovers the capabilities of any active interface. Network profile is then forwarded to the *Light Session Transfer Management (LSTM)* module. The LSTM is the heart of the whole architecture. It acts as a middleware between upper and lower layers, and receives network profiles and application profiles from the LNCD and the *Adaptation for Application (AfA)/Virtual Network Driver (VND)* modules, respectively. All information is stored in the *Storage*. The handover decision algorithm is processed in this module. After processing the decision algorithm LSTM notifies the AfA/VND module about the applications that should be moved to alternative interfaces. Components of the LSTM module are shown in Figure 3.

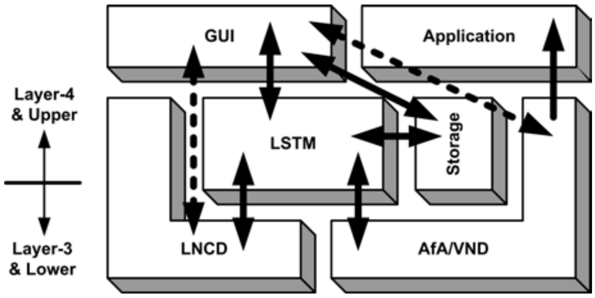

Figure 2. Stack diagram of the reference architecture

The AfA/VND module is adaptive to both Linux and Windows Mobile 2003 SE platforms. It can be assumed that the module contains two separate software parts, where the AfA works under Linux (as a previous version of the software) and the VND works under Windows Mobile 2003 SE. This module takes care of the session mobility. In this architecture no common mobility platform (e.g. Mobile IP) is present; instead, interface technology dependent individual and standard mobility features are used. It receives notifications from the LSTM module and accordingly shifts applications to alternative interfaces. In the process it takes care of the generation of new IP addresses and assigns them to applications. It also feeds the LSTM module with application profiles during the initialization phase. The GUI provides means to follow the complete processing of the architecture. It also enables users to configure the decision algorithm.

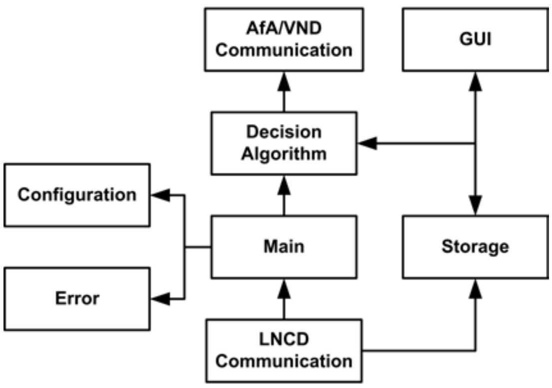

Figure 3. Components of the LSTM module

5. VALIDITY OF THE ALGORITHM

Test runs of the reference architecture software (with a web browsing application running on top) were performed in WLAN environment. Two APs acted as *dummy* access networks. Their capabilities were represented by individual *network capability configuration files* in the software, where appropriate file was identified by the *BSSID* of the corresponding AP. The test runs demonstrated that the decision algorithm worked perfectly and selected the best network intelligently. On the other hand, the user could influence the algorithm's behavior in an intuitive fashion by modifying the preferences. Thus, it gives us the confidence that the algorithm is capable of making intelligent decisions in accordance with user preferences and therefore, it should become a valuable aid for designers in designing smart software for future multimode terminals. Further simulation would be conducted in order to evaluate especially its performance.

6. CONCLUSIONS

In this paper, a context-aware decision algorithm based on the AHP method has been presented. The algorithm, which takes into account context information from both the terminal and network side, would be suitable for vertical HO decision making process in heterogeneous networks environment. The algorithm is fully flexible and dependent on the number of chosen objectives that will determine the dimension of the pairwise comparison matrix for objectives as well as the number of pairwise comparison matrices for networks in terms of each objective. The decision algorithm uses basic mathematical calculations that could be particularly suitable for embedded hardware in a mobile device. It is a service type based algorithm which means that the whole process is executed once for each type of running application, not for every running application. Thus, even in the worst case the total number of execution of the whole process is restricted to only three times while applications of all three types are running. This is particularly useful in minimizing processing time, handover delay, and CPU and memory usage.

Test runs of the reference architecture software demonstrated that the reference architecture as well as the decision algorithm would work perfectly in heterogeneous environment once the capabilities of the available networks are known.

7. REFERNCES

[1] T. L. Saaty, "How to make a decision: The Analytic Hierarchy Process", *European Journal of Operational Research*, 1990, Vol. 48, pp. 9-26.

[2] B. Hongyan, H. Chen, J. Lingge, "Intelligent Signal Processing of Mobility Management for Heterogeneous Networks", *Proc. of the 2003 IEEE Int. Conference on Neural Networks and Signal Processing*, 14-17 Dec. 2003, Vol. 2, pp. 1578-1581.

[3] P. Chan *et al.*, "Mobility Management incorporating Fuzzy Logic for a Heterogeneous IP Environment", *IEEE Communications Magazine*, Dec. 2001, Vol. 39, Issue 12, pp. 42-51.

[4] M. Stemm, R. H. Katz, "Vertical Handoffs in Wireless Overlay Networks", *ACM Journal on Mobile Networks and Applications*, Dec. 1998, Vol. 3, Issue 4, pp. 335-350.

[5] K. Pahlavan, et al., "Handoff in Hybrid Mobile Data Networks", *IEEE Personal Communications*, April 2000, Vol. 7, Issue 2, pp. 34-47.

[6] H. J. Wang, R. H. Katz, J. Giese, "Policy-Enabled Handoffs across Heterogeneous Wireless Networks", *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, 25-26 Feb. 1999, pp. 51-60.

[7] S. Balasubramaniam, J. Indulska, "Handovers between Heterogeneous Networks in Pervasive Systems", *IEEE Proc. on Communication Technology (ICCT 2003)*, April 2003.

[8] "Services and Service Capabilities", *3GPP TS 22.105 v. 3.6.0*, 1999.